CMSC 162
Introduction to Algorithmic Design II
Spring 2025

http://marmorstein.org/~robert/Spring2025/cs162.html

Lecture: 2:00 – 2:50pm MWF (Rotunda 354) **Lab:** 2:00 – 3:15pm T (Rotunda G54)

Instructor: Robert Marmorstein (marmorsteinrm@longwood.edu) **Phone:** 395-2185
Office Hours: 3:00-4:00pm MWRF or by appointment **Office:** Stevens 109

My schedule is posted near my office door. To make an appointment, please check the schedule to see which times I am free, then contact me by e-mail or Slack with your availability. In general, I need at least 24 hours of notice to schedule an appointment.

Course Description:
A continuation of CMSC 160. Topics include algorithmic design, complexity analysis, abstract data types, and encapsulation and basic data structures. Advanced topics using a modern high-level programming language include inheritance, overloading, and use of objects. 4 credits.

Prerequisite:
Grade of C- or better in CMSC 160

Prerequisite/Corequisite:
CMSC 161

Student Learning Outcomes:
At the end of this course, the successful student will be able to:

– apply object-based principles to creating understandable and maintainable solutions to problems

– identify appropriate implementations for abstract data types such as stacks, queues, lists, sets, trees, and maps

– explain, implement, and use data structures such as linked lists, trees, and hash tables

– compare and contrast standard algorithms using complexity analysis

Course Structure and Student Expectations
This is a heavily project-driven lecture course that combines theory with pragmatic programming practice. We will spend roughly three hours each week in lecture and discussion sections and one hour a week in class working on laboratory projects.

You should expect to spend at least nine additional hours each week outside of class reading the textbook, preparing for exams, and working on assignments.

Textbook and Other Resources:
The textbook for this class is "C++20 for Lazy Programmers", 2nd Edition, by Dr. Will Briggs, Apress, ISBN: 978-1484263051.

We will supplement this with readings from other sources such as the Unix Programmer's Manual (sometimes called the "man pages") and the TexInfo documentation (the "info" command). Additional readings will be posted on the course web site.

Course Requirements:
Your grade will depend largely on completion of the weekly lab sessions. These projects will comprise **50%** of your grade. The remainder of your grade will come from homework assignments and quizzes(**25%**), participation(**5%**), the midterm exam(**10%**), and the final exam(**10%**).

University Policies:
This course adheres to the university policies found at http://www.longwood.edu/academicaffairs/syllabus-statements/.

Grading Policy:
Your final grade in this course is computed using a weighted average of your scores on each assignment. The weights for each category are given in the course requirements section of this syllabus and can be used by applying the following formula:

Final Grade = 0.50 * Projects + 0.25 * (Homework and Quizzes) + 0.05 * Participation
+ 0.10 * Midterm + 0.10 * Final

Each of the category grades (such as Projects) can be computed by summing the points you've earned on each assignment in that category, multiplying by 100 and then dividing by the total number of points possible to obtain a percentage. Numeric grades are translated to letter grades using the following grading scale:

		100-91:	A	90:	A-
89:	B+	88-81:	B	80:	B-
79:	C+	78-71:	C	70:	C-
69:	D+	68-64:	D		

63 or lower: F

There is no grade of D- in this course. Anything below a 64 is failing.

Late Work:
In general, I do not accept late work or grant extensions on assignments unless you have a serious medical or family emergency which prevents you from completing the assignment on time (however, see "Slip days" below). In exceptional circumstances, I may be persuaded to grant extensions on one or more projects or assignments. In such cases, you do not need a doctor's note, but you must notify me of the circumstances within a reasonable amount of time (typically within twelve hours of the deadline).

All requests for extensions MUST be submitted by **e-mail** within a reasonable amount of time. This e-mail should outline in detail the reasons your work is late. Granting of extensions is entirely at my discretion – if you have not turned an assignment in on time, you should expect to earn a zero.

Slip Days:
You will be allocated a fixed number of slip days at the start of the semester. You may use your slip days to extend the due date of one or more *programming projects*. You can use all of your slip days on one assignment or you may use them over multiple assignments.

Slip days are calculated from the minute the assignment is due until you turn it in. The number of slip days used is rounded *up* to the nearest integer value. That means that if you turn an assignment in 24 hours and 1 minute after the due date, you will use up *two*

slip days. The slip day clock runs over weekends and holidays. If a lab is due on Friday and you turn it in on Monday, you will have used three slip days, not one. Slip days cannot be shared, traded, bought, or sold, but can occasionally be earned by participation in relevant campus activities I select.

Slip days may not be used on homework assignments, quizzes, or exams.

Attendance:
I expect you to attend class unless you are sick or engaged in a school-sponsored sport or extracurricular activity. Please do NOT come to class if you are sick. Instead, contact me within twelve hours of the absence to check whether you've missed any work and make arrangements to make up any missed quizzes. You should also make arrangements to get notes from another student in the class.

You should also check the course web site for announcements, new assignments, and other important updates.

I will rely primarily on your honor for enforcement of the attendance policy. However, I will keep a record of your attendance. In accordance with Longwood policy, missing more than 10% of scheduled class time (5 class sessions) to unexcused absences may, at my discretion, result in loss of one letter grade and missing 25% of class or more (14 sessions), whether excused or not may result in an automatic failing grade.

Cell Phones and Laptops:
Cell phones, music players, and laptops are to be turned off and put away during class, except as needed for the lab sessions. Violations of this policy will be considered an **unexcused** absence. I will not interrupt class to notify you if you have been counted absent for use of a prohibited device. Feel free to contact me by e-mail at any point in the semester to check on the number of absences you have in my class.

Food and Drink:
You may bring non-alcoholic beverages, including soft drinks, to class. However, please do not eat in class (it distracts me and the other students). Violations of this policy will be considered an **unexcused** absence. I will not interrupt class to notify you if you have been counted absent for violation of this policy. Feel free to contact me by e-mail at any point in the semester to check on the number of absences you have in my class.

I occasionally grant exceptions to this rule for students who must otherwise forgo lunch or have medical needs that require them to eat in class. If you feel that you need such an exception, you must make arrangements with me in advance (i.e. before bringing food to class).

Honor Code and Collaboration:
I firmly believe in the honor code. As such, I encourage you to actively collaborate with other students and to discuss homework problems. However, there is a point at which collaboration becomes cheating.

To help you understand the line between acceptable discussion of a project and dishonorable behavior, I ask you to observe the following rules:

1. Exams and quizzes are to be completed entirely on your own. You may not discuss them with anyone or use any resources except those specifically outlined on the exam handout.

2. You must give proper attribution.

Whenever you receive help or use an online resource, you should comment your code to give proper credit. The best way to do this is to place a comment ***above or on the same line*** *as the code on which you received help or used a resource. For example:*

/ based on http://codewarrior.com */*

or

/* Jessica helped me with the curly braces here */

is fine. You **DO NOT** need to cite material you obtain directly from me (in lecture, the assignment handout, or office hours). In general, you also **DO NOT** need to cite material taken from the textbook.

3. The work you submit should, in general, be either your own original work or material which I have provided and you have suitably modified yourself.

You **MAY** use web sites, books, and the man pages as reference materials. However, you must cite them appropriately and you **MUST** re-type any code you find and not just download it or copy/paste it.

At no point should another student touch your keyboard while helping you with a project.

For homework and projects, everything you turn in should be something YOU have personally typed or hand-written. You may NOT copy code electronically from other students or the Internet.

You **MAY NOT** share code with other students using flash drives, cell phones, e-mail, web sites, floppies, CDs, or other means unless I specifically direct you to do so. You **MAY NOT** print out copies of your code to share with other students (personal copies or copies to show me during office hours are fine).

4. Do not copy large blocks of code from other students or the Internet.

You **MAY** assist other students or get assistance with simple problems like syntax errors, but you **MAY NOT** copy large blocks of code, such as entire classes or functions, from a web site or from another student. How much code is “too much” depends partly on context, but a good guideline of what "large" means is that copying more than three complete programming statements is usually too much.

5. You may not use Generative AI on your assignments.

Generative AI technologies (including large language models, such as ChatGPT, Google Bard/Duo, Meta’s Llama, and Github Co-pilot) use sources without proper attribution. As such, use of these tools implicitly commits plagiarism, an honor code violation. You MAY NOT use these tools in any way on work you submit for this course. You MAY use these tools to generate examples for your own learning and study as long as none of the code or other content you generate is used (even accidentally) on any submitted work.

6. You are responsible for securing your code.

Helping other students to cheat is also cheating. Furthermore, it is your responsibility to make sure that other students do not use your work to cheat. Be careful with who you let

access your account and report any missing files, flash drives, or other devices to me promptly.

Infractions of these policies will be dealt with harshly under the Longwood Honor Code. Any student convicted of an honor offense involving this class will automatically receive a final course grade of **F** in addition to any penalties imposed by the Honor Board. You should consider all work in this class to be pledged work, whether or not the pledge appears on the assignment.

If you have questions about the honor code policy, PLEASE ask me. It is much better to receive a late penalty on a single assignment than to fail the course and face honor board charges.

You may find the scenarios at https://integrity.mit.edu/handbook/writing-code helpful in understanding this policy. While their honor code policy is not identical to mine it is similar.

Major Assignments:

In addition to the two exams, there will be homework assignments, quizzes, and several laboratory projects in this course.

Projects:

Projects are worth 50% of your grade. There will be six to eight laboratory projects. For tentative due dates, see the course schedule on this syllabus.

Exams:

The midterm exam will be taken in class on March 7th and is worth 10% of your final grade. The final exam will be held on Monday, May 6th at 3:00pm and is also worth 10% of your grade.

Homework Assignments and Quizzes:

These will largely be drill worksheets intended to help you practice and retain key concepts from the course. They will be worth a varying, but usually small, number of points. In addition, I will provide you with review materials for the two exams. These review packets will count toward the homework grade and will be worth a significant number of points.

I give unannounced pop quizzes. These are usually worth 10 to 20 points and count toward the homework grade. Together, homework assignments and quizzes will make up 25% of your final course grade.

Communications Policy:

The best way to get in touch with me is to use **Slack**. Slack is a chat utility with clients for mobile devices and desktop computers. I recommend you install it on both types of devices. Slack will allow you to easily send me code snippets, ask questions in real time, or set up a Zoom meeting if we need to video chat. You should sign up for a Slack account by visiting https://longwood-cmsc.slack.com. Use your @live.longwood.edu email address to register and you will be automatically approved for an account.

I will expect you to check the **#cmsc-161** channel every day before class in case I have posted an announcement or asked you to bring something to class.

When you send me a Slack message, I instantly get a notification on my computer, tablet, and phone. Typically, I will reply to Slack messages within 24 hours (often sooner) on weekdays. While I am often available in the evening or on weekends, you may need to be patient if I am busy with other students or family obligations.

If you are **asking for help with a project or homework problem**, you should attach your work to a direct message in Slack so that I can see where you are at. You should do this by using the "plus" icon to attach the file directly to your message or by copy/pasting the particular snippet of code you are working on to the body of the message.

Please do NOT attach pictures of your work taken on your phone. These are often blurry and always hard to read. Also, if you attach your code I can run it to see why it is failing, but if you only send me a picture of it, I will have to "guess" why it is wrong. Nevertheless, it is sometimes useful to be able to see a picture of your screen. The best way to do this is to take a screenshot of your system using the "Spectacle" program (usually by pressing the Print Screen "PrtSc" key).

One last suggestion: **don't "ask to ask"**. Asking me whether you can ask a question wastes my time and yours. I am delighted to answer questions about the projects and homework assignments and you should feel free to ask questions at any time (yes, even 3am the night before the project is due – I MIGHT be awake and online – and I'm happy to help you find an answer).

Slack is also a good way to communicate with other members of the class. You will be invited to a public **#cmsc-161** channel in which you can discuss the projects and other course topics with other students in the class. Feel free to ask for help on this channel, but please stick to general questions rather than posting code.

You can also reach out to me by e-mail to marmorsteinrm@longwood.edu. However, please do not send me large files by e-mail. They take up space toward my limited quota on the mail server and cause me all sorts of headaches. **E-mail messages containing large files will be deleted unread**.

I am much slower at replying to e-mail (since I do not get a notification and have to log in to check it). Typically, you can expect a reply to an e-mail within 48 hours, but this may be longer on weekends, and I may not receive your message at all or may not be able to respond to it (my inbox is often over the "quota" allowed by campus I. T. and this often prevents me from using the system effectively).

Tentative Course Schedule:

Week 1: Jan. 15 – 17

Introduction and Review
Working in a Unix Environment

Review of Variables, Loops, and Functions
Read Syllabus
Read Chapter 1

Jan. 20 Dr. Martin Luther King Jr Day: NO CLASS

Jan. 21 Lab 0: C++ Review and Introduction to SSDL (Due Jan. 27)

Week 2: Jan. 21 – 24

Review of Files, File Streams, and String Streams
Review of Enumerations, Arrays, and Vectors

Searching: Linear and Binary Search

Structs

Read Chapters 10 and 11 of C++20 for Lazy Programmers

Jan. 23 Last day of Add/Drop (by 5 pm)

Week 3: Jan. 27 – 31

Dynamic Memory, Pointers, LValues, Rvalues, and References
Smart Pointers

Dynamic Arrays
Libraries and Linking

Read Chapters 13 and 14

Jan. 28 Lab 1: Files, Streams, and Vectors (Due: Feb. 3)

Week 4: Feb. 3 – 7

Classes, Constructors and Destructors, Initializer Lists
Design Principle: Code Reuse and Modularity
Testing and Debugging

Mutators and Accessors, Encapsulation
Implementation and Interface
Constants and Mutability
In-line Functions

Read Chapters 15 and 16

Feb. 4 Lab 2: Classes and Objects (Due: Feb. 17)

Week 5: Feb. 10 – 14

Operators and Exceptions
Operator Overloading

Read Chapter 17

Feb. 11 Project Work Day

Week 6: Feb. 17 – 21

Move Semantics, Exceptions, and Recursion

std::swap and std::move
Big-O Analysis
Static Members
Internal and External Documentation

Read Chapter 18

Feb. 18 Lab 3: Streams and Vectors (Due: Mar. 24)

Week 7: Feb. 24 – 28

Inheritance
Abstraction and Polymorphism
Design Principle: Top-Down Design

Read Chapter 19

Feb. 25 Project Work Day

Week 8: Mar. 3 – 7 — Templates, Multiple Inheritance, Virtual and Pure Virtual Functions
Read Chapters 20 and 21

Midterm Exam (Mar. 7th)

Mar. 4 **Exam Review**

Mar. 10 – 14 **Spring Break: NO CLASS**

Week 9: Mar. 17 – 21 — Linked Lists, Array and Pointer based Lists, Abstract Data Types

Stacks and Queues
Read Chapter 22

Mar. 18 **Lab 4: Linked Lists, Stacks and Queues (Due: Mar. 24)**

Week 10: Mar. 24 – 28 — Iterators, Casting, and Type Deduction
C++ Concepts
Read Chapter 23

Mar. 25 **Lab 5: Stacks, Graphics, and Exceptions (Due: Apr. 7)**

Week 11: Mar. 31 – Apr. 4 — Recursion, Trees
Tree Traversals
Balanced Binary Search Trees

Apr. 1 **Project Work Day**

Apr. 2 **Deadline to withdraw without an 'F'**

Week 12: Apr. 7 – 11 — Priority Queues / Heaps
Smart Pointers

Read Chapter 26

Apr. 8 **Lab 6: Hashing and using an IDE (Due: Apr. 21)**

Week 13: Apr. 14 – 18 — Hash Functions, and Hash Tables, Open and Closed Hashing
Using static analysis tools

Apr. 15 **Project Work Day**

Week 14: Apr. 21 – Apr. 25 — Sorting: Selection Sort, Bubble Sort Insertion Sort

Apr. 22 **Lab 7: Sorting (Due: Apr. 28)**

Apr. 23 **Research Day: NO CLASS**

Week 15: Apr. 28 – May 2 — Advanced Sorting: Quick Sort, Merge Sort, and Heap Sort

Apr. 29 **Exam Review**

May 6 **Final Exam (Tuesday, 3:00 – 5:30pm)**